A Compact Routing Scheme and Approximate Distance Oracle for Power-law Graphs[1]

WEI CHEN, Microsoft Research Asia
CHRISTIAN SOMMER, Massachusetts Institute of Technology
SHANG-HUA TENG, University of Southern California
YAJUN WANG, Microsoft Research Asia

Compact routing addresses the tradeoff between table sizes and stretch, which is the worst-case ratio between the length of the path a packet is routed through by the scheme and the length of an actual shortest path from source to destination. We adapt the compact routing scheme by Thorup and Zwick (SPAA 2001) to optimize it for power-law graphs. We analyze our adapted routing scheme based on the theory of unweighted random power-law graphs with fixed expected degree sequence by Aiello, Chung, and Lu (STOC 2000). Our result is the first analytical bound coupled to the parameter of the power-law graph model for a compact routing scheme.

Let n denote the number of nodes in the network. We provide a *labeled* routing scheme that, after a stretch–5 handshaking step (similar to DNS lookup in TCP/IP), routes messages along stretch–3 paths. We prove that, instead of routing tables with $\tilde{O}(n^{1/2})$ bits ($\tilde{O}$ suppresses factors logarithmic in n) as in the general scheme by Thorup and Zwick, expected sizes of $O(n^{\gamma} \log n)$ bits are sufficient, and that all the routing tables can be constructed at once in expected time $O(n^{1+\gamma} \log n)$, with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$, where $\tau \in (2, 3)$ is the power-law exponent and $\varepsilon > 0$ (which implies $\varepsilon < \gamma < 1/3 + \varepsilon$). Both bounds also hold with probability at least $1 - 1/n$ (independent of ε). The routing scheme is a labeled scheme, requiring a stretch–5 handshaking step. The scheme uses addresses and message headers with $O(\log n \log \log n)$ bits, with probability at least $1 - o(1)$. We further demonstrate the effectiveness of our scheme by simulations on real-world graphs as well as synthetic power-law graphs.

With the same techniques as for the compact routing scheme, we also adapt the approximate distance oracle by Thorup and Zwick (STOC 2001, J.ACM 2004) for stretch 3 and we obtain a new upper bound of expected $\tilde{O}(n^{1+\gamma})$ for space and preprocessing for random power-law graphs. Our distance oracle is the first one optimized for power-law graphs. Furthermore, we provide a linear-space data structure that can answer 5–approximate distance queries in time at most $\tilde{O}(n^{1/4+\varepsilon})$ (similar to γ, the exponent actually depends on τ and lies between ε and $1/4 + \varepsilon$).

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph Theory; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Algorithms, Theory

Additional Key Words and Phrases: shortest path, shortest-path query, distance oracle, complex networks, power-law graphs

[1]This work has been announced at the 23rd International Symposium on Distributed Computing (DISC 2009); an extended abstract appeared in the corresponding proceedings on pp. 379–391.

Work by C. Sommer partially supported by the Swiss National Science Foundation.
Author's addresses: C. Sommer, Massachusetts Institute of Technology, Cambridge, MA, corresponding author, csom@csail.mit.edu; W. Chen and Y. Wang, Microsoft Research Asia, Beijing, China; S.-H. Teng, University of Southern California, Los Angeles, CA.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

ACM Reference Format:
Chen, W., Sommer, C., Teng, S.-H., Wang, Y. 2012. A Compact Routing Scheme and Approximate Distance Oracle for Power-law Graphs. ACM Trans. Alg. ?, ?, Article ? (January 201?), 26 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Message routing and answering shortest-path and distance queries are fundamental services for communication and information networks. Routing can be seen as the distributed version of answering a shortest-path query. When answering a distance query for a pair of nodes in a network, an efficient algorithm accesses only a small fraction of the information stored at preprocessing time. A preprocessing algorithm, the data structure it creates, and its corresponding query algorithm that computes (approximate) shortest distances, are referred to by the term *distance oracle*. When routing a message from a source to a destination in the network, to decide where to forward the message to, a node may only use its local information, which includes its local routing table, the destination address, and a message header.

Both routing schemes and distance oracles are expected to "produce" (route messages or output) shortest or approximate shortest paths between all source-destination pairs along. A key measure for the quality of routing schemes (and distance oracles) is their worst-case multiplicative *stretch*, which is defined as the maximum ratio of the length of the message route between a pair of nodes s and t by the scheme and the actual shortest path length between s and t, among all s-t pairs in the network.

Routing schemes address the tradeoff between stretch and routing table size. A trivial stretch-1 routing scheme is one in which every node stores for every destination in the network where to forward the message to. However, for a network with n nodes, this approach requires unscalable $\Omega(n \log n)$-bit routing tables for every node [Gavoille and Perennes 1996]. A *compact* routing scheme is only allowed to have routing tables with sizes sublinear in n and message header sizes polylogarithmic in n. There are two classes of compact routing schemes: *Labeled* schemes are allowed to add labels to node addresses to encode useful information for routing purposes, where each label has length at most polylogarithmic in n. *Name-independent* schemes do not allow the renaming of node addresses, instead they must function with all possible addresses.

1.1. Our Contributions

We bridge the gap between theory and practice in the study of compact routing schemes and distance oracles for *power-law graphs*. In a power-law graph (sometimes also termed *scale-free* network or graph), the number of nodes with degree x is proportional to $x^{-\tau}$, for some constant τ, often between 2 and 3. Experimental results [Krioukov et al. 2004] suggest that there are efficient routing schemes for power-law graphs. We provide the first theoretical analysis that directly links the power-law exponent τ of a random power-law graph to the bound on the routing table sizes (and the distance oracle space complexity, respectively).

More specifically, we adapt the labeled universal compact routing scheme of Thorup and Zwick [2001] to optimize it for unweighted, undirected power-law graphs. Our adaptations include (a) selecting nodes with the largest degrees as the landmarks instead of random sampling, and (b) directly encoding shortest paths in node labels and message headers instead of relying on a tree routing scheme. The details of the scheme can be found in Section 4; a detailed comparison with [Thorup and Zwick 2001] is deferred to Section 2.3.

Our complexity analysis (see Section 5) of the routing scheme is based on the random power-law graph model with expected degree sequence proposed by Aiello, Chung and Lu [Aiello et al. 2000; Chung and Lu 2002; Chung and Lu 2006; Lu 2002b] with some modifications. We assume the power-law exponent τ to lie in the range of $(2, 3)$, which is the so called "finite mean infinite variance" region of the power-law degree distribution, where most practical power-law networks are assumed to be in [Clauset et al. 2009, p. 662].

We prove that for a stretch upper bound of 3 (after a handshaking step with stretch 5), instead of tables of size $\tilde{O}(n^{1/2})$ shown to be optimal up to a polylogarithmic factor for general graphs [Thorup and Zwick 2001], expected sizes of $O(n^{\gamma} \log n)$ bits are sufficient, and that the routing tables can be constructed at once in expected time $O(n^{1+\gamma} \log n)$, with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ and $\varepsilon > 0$ (which implies $\varepsilon < \gamma < 1/3 + \varepsilon$). Both bounds also hold with probability at least $1 - 1/n$ (independent of ε). This means that for all $\tau \in (2, 3)$, we have an upper bound of $\tilde{O}(n^{1/3+\varepsilon})$ on the routing table sizes, which is better than the optimal bound of $\tilde{O}(n^{1/2})$ for general graphs. For values of τ close to 2, for example for $\tau = 2.1$, our bound is $O(n^{1/12+\varepsilon})$. The routing scheme requires a stretch–5 *handshaking step*, which is similar to DNS lookup in TCP/IP (see also [Thorup and Zwick 2001, Sec. 4]). Thorup and Zwick also point out that labeled routing schemes probably require a lookup step anyway and that one stretch–5 step may be negligible for communication patterns that span several messages. Our scheme uses addresses and message headers of size $O(\log n \log \log n)$, with probability at least $1 - o(1)$. The efficient encoding using $O(\log n \log \log n)$ bits in addresses and headers relies on specific distance properties of power-law graphs. Our scheme is a *fixed-port* scheme, meaning that it works for any permutation of port number assignments on any node.

We provide simulation results for both random power-law graphs and actual router-level networks, which demonstrate the effectiveness of our adapted compact routing scheme (Section 6).

Using the same techniques, we also adapt the approximate distance oracle by Thorup and Zwick [2005] for unweighted, undirected power-law graphs (see Section 7 for details). We prove that, for stretch 3, instead of an oracle of size $O(n^{3/2})$, expected space $O(n^{1+\gamma})$ is sufficient and that the oracle can be constructed in expected time $O(n^{1+\gamma} \log n)$. Again, both bounds also hold with probability at least $1 - 1/n$. Furthermore, we provide a linear-space data structure that can answer 5–approximate distance queries in time at most $\tilde{O}(n^{1/4+\varepsilon})$ (similar to γ, the exponent actually depends on τ and lies between ε and $1/4 + \varepsilon$).

1.2. Organization

We first outline related work on power-law graphs (Section 2.1), (compact) routing schemes (Section 2.2), including a detailed comparison with the routing schemes by Thorup and Zwick (Section 2.3), and approximate distance oracles (Section 2.4). Our routing scheme and distance oracles are for a certain model of random power-law graphs, which we define in Section 3. The compact routing scheme for random power-law graphs is described in Section 4; we analyze its theoretical performance in Section 5 and its experimental performance in Section 6. We provide and analyze approximate distance oracles for random power-law graphs in Section 7. We conclude with a summary and open questions in Section 8.

2. RELATED WORK

2.1. Power-law Graphs

Power-law graphs [Mitzenmacher 2003] constitute an important family of networks appearing in various real-world scenarios such as social networks and many more are claimed to be power-law graphs [Clauset et al. 2009; Faloutsos et al. 1999], sometimes rather controversially [Achlioptas et al. 2009; Roughan et al. 2011]. In a power-law graph, the number of nodes with degree x is proportional to $x^{-\tau}$, for some constant τ. The power-law exponent τ for many real-world networks is in the range between 2 and 3 [Krioukov et al. 2004, Sec. I.B]. Power-law graphs do not seem to belong to any of the well-studied network families such as trees, planar graphs or low-doubling-dimension graphs (see Section 2 for details). Despite their unique features, power-law graphs are actually not "easy" instances for algorithms. Although power-law graphs are sparse, optimization problems remain hard: problems such as COLORING or CLIQUE are **NP**-hard for power-law graphs as well [Ferrante et al. 2008].

Besides the random power-law graph model of Aiello, Chung, and Lu [Aiello et al. 2000; Chung and Lu 2002; Chung and Lu 2006; Lu 2002b], other mathematical models for power-law graphs include the configuration model [Newman et al. 2001], the Poissonian process [Norros and Reittu 2006], and the preferential attachment model [Barabási and Albert 1999; Kumar et al. 2000]. Among these, the random power-law graph model is studied very well, providing a rich body of mathematical results. Furthermore, recent empirical studies on compact routing also use this model [Brady and Cowen 2006; Krioukov et al. 2004].

2.2. (Compact) Routing Schemes

Both labeled and name-independent compact routing schemes have been studied extensively. Universal schemes work for all network topologies [Abraham et al. 2006b; Abraham et al. 2006c; Abraham et al. 2008; Cowen 2001; Peleg and Upfal 1989; Thorup and Zwick 2001]. It has been shown that with $\tilde{O}(n^{1/k})$-bit routing tables (as usual, we abbreviate $O(f(n) \cdot \log^t n)$ for some constant t by $\tilde{O}(f(n))$) one can achieve a stretch of $O(k)$, and that this tradeoff is essentially tight due to a girth conjecture by Erdős.

Due to these impeding lower bounds for general graphs, specialized schemes were designed for various families of network topologies, including trees [Fraigniaud and Gavoille 2001; Korman 2008; Thorup and Zwick 2001], planar graphs [Gavoille and Hanusse 1999; Lu 2002a], fixed-minor-free graphs [Abraham et al. 2005], or graphs with low doubling dimension [Abraham et al. 2006a; Konjevod et al. 2006; Konjevod et al. 2007]. These topology-specific schemes achieve significant improvements on the stretch–space tradeoff over universal routing schemes.

Despite their potential relevance in practice, the family of power-law graphs has not received much attention from the routing research community. There are experimental studies of compact routing in power-law graphs and Internet-like graphs. Krioukov et al. [2004] evaluate the universal routing scheme of Thorup and Zwick (*TZ*) [2001] on random power-law graphs [Aiello et al. 2000] and provide experimental evidence of much better performance (both in terms of stretch and table sizes) than the theoretical worst-case bound. However, they do not provide a analytical bound of the TZ scheme on power-law graphs for neither stretch nor table size. Enahescu et al. [2008] propose a *greedy* landmark selection scheme and they show empirically that their adaptation achieves good stretch and table sizes for power-law graphs and Internet Autonomous System (AS) graphs. Unfortunately, their theoretical analysis is for Erdős-Rényi random graphs [Erdős and Rényi 1960] instead of power-law graphs. Brady and Cowen [2006] give a compact routing scheme tailored for power-law graphs with additive stretch d and header and table sizes $O(e \log^2 n)$, where both d and e depend on the graph, and they show experimentally that these values are reasonably small for certain random power-law graphs [Aiello et al. 2000]. However, there is no rigorous analysis connecting d and e to the parameter τ of power-law graphs.

Embedding power-law graphs into *hyperbolic spaces* and using the coordinates appears to be a promising approach for routing in power-law graphs [Cvetkovski and

Crovella 2009; Papadopoulos et al. 2010]. However, current approaches do not offer any guarantees on the worst-case stretch and, furthermore, both success probability and stretch have only been evaluated experimentally so far.

2.3. Detailed Comparison with Existing Routing Schemes

Thorup and Zwick's routing schemes. We first provide additional details on the comparison with Thorup and Zwick's routing schemes.

Thorup and Zwick [2001] contribute two different routing schemes. Their first scheme is a stretch–3 scheme with an $O(n^{1/2}\log^{3/2} n)$-bit routing table per node and $O(\log n)$-bit labels and headers. This scheme is based on Cowen's earlier scheme [2001], which uses a small subset A of nodes, called *landmarks*, to route messages. In a graph $G = (V, E)$ with landmark set $A \subseteq V$, for every node u, define its *cluster* $C(u) = \{v \in V \ : \ d(v, u) < d(v, A)\}$, where $d(v, u)$ and $d(v, A)$ denote the graph distance from v to u and A, respectively. Let $\ell(u)$ denote the landmark in A that is the closest to node u (ties are resolved arbitrarily). The routing table of node u stores the port identifiers to route messages to all nodes in A and $C(u)$. If a destination v is not in $A \cup C(u)$, u routes through $\ell(v)$, which guarantees a stretch bound of 3 due to the definition of the cluster $C(u)$. Thorup and Zwick use a resampling method to achieve $|A \cup C(u)| = O(n^{1/2}\log^{1/2} n)$ for every node u. It may be tempting to adapt TZ's first scheme (as described above) for random power-law graphs. Our analysis, however, breaks down due to dependency issues (see Section 5.7 for more details).

The second scheme of Thorup and Zwick [2001] is based on their approximate distance oracle [Thorup and Zwick 2005]. For any $k \geq 2$, they design a compact routing scheme with $\tilde{O}(n^{1/k})$-bit tables, $O(k \log^2 n/ \log\log n)$-bit addresses, and $O(\log^2 n/ \log\log n)$-bit headers (the bounds on addresses and headers are for fixed-port schemes). The scheme achieves stretch $2k - 1$ with a stretch $4k - 5$ handshake. For the case of $k = 2$ (comparable to our scheme), their scheme essentially considers the landmark set A together with the *ball* of a node u, $B(u) = \{v \ : \ d(v, u) < d(u, A)\}$. Note that balls and clusters are dual concepts: $v \in C(u)$ if and only if $u \in B(v)$. The routing table of u stores the ports to route messages to all nodes in $A \cup B(u)$. Similar to the first scheme, when $v \notin A \cup B(u)$, u routes through $\ell(v)$ to reach v, but in this case it only guarantees a stretch of 5 instead of 3 when $v \notin B(u)$ but $u \in B(v)$. A *handshake* is needed to reduce the stretch to 3. Moreover, a node w on the path from $\ell(v)$ to v may not know the port to route to v from its routing table, since v may not be in $B(w)$ (though $v \in C(w)$). To resolve this issue, Thorup and Zwick further use a tree routing scheme, which requires additional, rather complicated labels. They use random sampling to guarantee that $|A \cup B(u)| = \tilde{O}(n^{1/2})$.

Our scheme is similar to their second scheme. We also use balls and landmarks to route messages. There are two major differences: First, we use high-degree nodes instead of randomly selected nodes as landmarks. The major contribution of the paper is to prove that, with this selection strategy, in random power-law graphs, we achieve $|A \cup B(u)| = O(n^{\gamma})$ with $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ and $\varepsilon > 0$, which holds both in expectation and with high probability. Second, instead of using a tree routing scheme, we directly encode the shortest path from $\ell(v)$ to v in v's address, which is short (with probability $1 - o(1)$) due to the distance properties in random power-law graphs. As a result, our routing table sizes are smaller than the tables in both TZ schemes, and our address and header size of $O(\log n \log\log n)$ is better than TZ's second scheme and close to TZ's first scheme. Our scheme is also simpler than the second scheme and is comparable with the first scheme. This improvement is possible only by tailoring the scheme to unweighted power-law graphs.

Probabilistic guarantees. Recently, there have been compact routing schemes with probabilistic guarantees. Since for our scheme also some properties also hold with high probability, let us briefly describe the differences to existing schemes. Routing schemes by Dinitz [2007] can efficiently find short routes for most pairs, there is however some *slack*, meaning that for some pairs the route may be longer. In our work, the bound on the *stretch* is for the worst-case pair $(s, t) \in V \times V$. In the Dinitz scheme, the *slack* is with respect to the stretch: for some pairs (constant fraction) the stretch inequalities do not necessarily hold. In our work, the bound on the *table size* holds with high probability $(1 - 1/n)$. Our construction is deterministic but analyzed in an average-case manner for a specific class of random graphs. The Thorup-Zwick construction is randomized and works for any graph. Hence, we both obtain a bound on the *table size* with high probability.

2.4. Distance Oracles

Dijkstra's algorithm [1959] finds a shortest path in any graph with non-negative edge weights in time $O(n \log n + m)$, where n and m denote the number of nodes and edges, respectively. For applications such as navigation software exploring huge maps or for social networking sites, this query time is not practical. Instead, the graph is *preprocessed* and a special data structure allows for efficient *queries*. One way to prepare for queries is to precompute all shortest paths using an All-Pairs Shortest Path algorithm [Chan 2007] and to read a shortest path from a distance table. Time and memory constraints, however, render this approach impractical. Instead of running a cubic-time algorithm and using quadratic storage, we want to efficiently preprocess a graph to allow for fast distance queries. However, for general (directed) graphs with n vertices, $\Omega(n^2)$ space is necessary to return the shortest distance. *Approximate distance oracles* address the trade-off between approximation ratio, space, and preprocessing and query time, and can thus be interpreted as a generalization of the All-Pairs (Approximate) Shortest Path problem. In the following, we list approaches tailored for complex networks. For recent surveys we refer to [Sen 2009; Sommer 2010].

Thorup and Zwick [2005] provide a stretch–$(2k-1)$ distance oracle of size $\tilde{O}(kn^{1+1/k})$, which can be constructed in time $O(kmn^{1/k})$. Assuming a girth conjecture by Erdős, stretch and size are asymptotically tight for small values of k. For stretch parameter $k = 2$, the distance oracle of Thorup and Zwick has the following worst-case performance: the size is $O(n^{3/2})$ and the stretch is 3. Fortunately, the theoretical worst-case stretch bounds of Thorup and Zwick's distance oracle [2005] (and, also, of their routing scheme [Thorup and Zwick 2001]) are not observed in experiments [Krioukov et al. 2004], even though they are tight.

Distance oracles and shortest-path queries for complex networks have been studied experimentally. Potamias et al. [2009] use *landmark-based A* search* [Goldberg and Harrelson 2005]. Das Sarma et al. [2010] provide a practical implementation of Bourgain's embedding [1985], and they propose an extension of the distance oracle by Thorup and Zwick [2005]. In their extension, they omit ball computations. While the asymptotic performance is not affected, their algorithms both for preprocessing and query are simpler and potentially faster in practice than the corresponding original algorithms. The stretch bounds, however, only hold with high probability. Cheng and Yu [2009] use *2–hop labels* [Cohen et al. 2003] to efficiently compute exact distances. Xiao et al. [2009] compress graphs by exploiting symmetries. Instead of treating vertices as a single unit, they work on *orbits* of *automorphism groups*. Shortest-path queries are answered using *compact BFS-trees*, which are based on these orbits. Symmetries in complex networks seem to be very common. Experiments show that their method may be very efficient; the running time of the preprocessing algorithm appears to be roughly quadratic in the number of nodes. Goldman et al. [1998] consider relationships among objects in large databases. Their method processes keyword searches over databases in interactive query sessions. Distances between objects are computed based on a compact index, which consists of local neighborhoods and distances to *hub* vertices (separators). Hubs are chosen as high-degree nodes.

Although complex networks seem to be rather common in practice, to the best of our knowledge, there is no distance oracle with provable guarantees better than those of the general distance oracle of Thorup and Zwick [2005].

3. PRELIMINARIES

We adapt the random graph model for fixed expected degree sequence as defined by Aiello, Chung, and Lu [Aiello et al. 2000; Chung and Lu 2002; Lu 2002b; Chung and Lu 2006] using the definition from [Chung and Lu 2002, Section 2]. We refer to the original random graph distribution using the expression Fixed Degree Random Graph (**FDRG**).

Definition 3.1 (*Fixed Degree Random Graph [Chung and Lu 2002, Section 2]*). In a random graph with a given *expected degree sequence* $\vec{w} = \{w_1, w_2, \dots, w_n\}$ such that $\forall i : w_i^2 < \sum_j w_j$, the edge between v_i and $v_{i'}$ is present in the random graph with probability

$$\Pr[\{v_i, v_{i'}\} \in E] = w_i w_{i'} \rho, \text{ where } \rho = \frac{1}{\sum_j w_j}.$$

In the original **FDRG** model it is assumed that $\forall i, i' : w_i w_{i'} < \sum_j w_j$. We adapt the original model by deterministically inserting edges if $w_i w_{i'} > \sum_j w_j$. Without modification, the original assumption would rule out the values for τ considered in this work.

Definition 3.2. For a constant $\tau \in (2, 3)$, the random power-law graph distribution $\mathbf{RPLG}(n, \tau)$ is defined as follows. Let the sequence of generating parameters $\vec{w} = \{w_1, w_2, \dots, w_n\}$ obey a power law:

$$w_j = \left(\frac{n}{j}\right)^{1/(\tau-1)} \text{ for } j \in \{1, 2, \dots n\}.$$

The edge between v_i and $v_{i'}$ is present in the random graph with probability

$$\Pr[\{v_i, v_{i'}\} \in E] = \min\{w_i w_{i'} \rho, 1\}, \text{ where } \rho = \frac{1}{\sum_j w_j}.$$

Note that, in both models, there is a one-to-one correspondence between a node v_j and its generating parameter w_j. In the **FDRG** model, the value w_j corresponds to the expected degree of vertex v_j, and Chung and Lu refer to $\vec{w}$ as the *expected degree sequence*. In the $\mathbf{RPLG}(n, \tau)$ adaptation, the graph is sampled according to the *generating parameter values* w_j. Let D_j be the random variable denoting the degree of node v_j. In the $\mathbf{RPLG}(n, \tau)$ model, the expected degree $E[D_j]$ of node v_j is less than or equal to the generating parameter w_j. We refer to the edges between two nodes $v_i, v_{i'}$ with $w_i w_{i'} \geq \sum_j w_j$ as *deterministic edges*; we refer to the remaining edges as *random edges*.

An important technical reason to work with the model of Aiello, Chung, and Lu [Aiello et al. 2000; Chung and Lu 2002; Lu 2002b; Chung and Lu 2006] is that the edges are independent. This independence makes several graph properties easier to analyze. We also (implicitly) rely on a property called *assortativity*. Assortativity is the tendency of nodes with high degree to attach to other highly connected nodes. This tendency is especially high in social networks. The opposite tendency, termed *dissortativity*, is more common in technological and biological networks, wherein highly connected nodes tend to be connected with low-degree nodes. Intuitively speaking, the property we rely on is that a node with high degree is either in the so-called *core* (as defined below) or, with high probability, at least one of its neighbors is in the core. Li et al. [2005, Definition 4.1] formalize assortativity as follows. They define the $s(G)$ value of a graph as $s(G) := \sum_{\{v_i,v_{i'}\}\in E} \deg(v_i)\cdot\deg(v_{i'})$. Graphs sampled from the **FDRG** model tend to have a high $s(G)$ value, since high-degree nodes are attached to other highly connected nodes. Li et al. state that $s(G)$ measures to what extent a graph has a "hub-like core." The *core* of a graph consists of nodes having large degrees.

Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ for some $\varepsilon > 0$ and $\gamma' = \frac{1-\gamma}{\tau-1}$. We require that $n = |V(G)|$ is sufficiently large, specifically, that

$$n^{\frac{\varepsilon(2\tau-3)}{\tau-1}} \geq \frac{2(\tau-1)}{\tau-2}\ln n. \tag{1}$$

Our results do not have any other implicit dependencies on ε.

Definition 3.3. For a power-law degree sequence $\vec{w}$ and a graph G with n nodes, the core with degree threshold $n^{\gamma'}$, $\gamma' \in (0,1)$, is defined as follows.

$$\begin{aligned}\mathsf{core}_{\gamma'}(\vec{w}) &:= \{v_i : w_i > n^{\gamma'}\},\\ \mathsf{core}_{\gamma'}(G) &:= \{v_i : \deg_G(v_i) > n^{\gamma'}/4\},\end{aligned}$$

where $\deg_G(v_i)$ is the degree of v_i in G (the subscript G is omitted when the graph is clear from the context).

Our $\mathsf{core}_{\gamma'}(\vec{w})$ is the $n^{\gamma'}$-Core in [Lu 2002b, Chapter 4, Definition 2].

For each vertex u of a graph G, we define its ball relative to the core as

$$B_G(u) := \{v \in V(G) : d(u,v) < \min_{v'\in\mathsf{core}_{\gamma'}(G)} d(u,v')\}.$$

4. THE ADAPTED COMPACT ROUTING SCHEME

Let the unweighted graph $G = (V,E)$ model the network. Each node v in the network has a unique $\lceil \log_2 n\rceil$-bit static name. Whenever we write v in a routing table, a message header, or a node address, we mean its $\lceil \log_2 n\rceil$-bit static name representation. Each node v has $\deg(v)$ ports connecting it with its neighbors. These ports are numbered by $0,1,\ldots,\deg(v)-1$, and identifying each port number of v requires $\lceil \log_2 \deg(v)\rceil$ bits. For every packet, the routing scheme needs to decide which port the packet is to be forwarded to. Our scheme is a fixed-port scheme, that is, it works with arbitrary permutations of port number assignments.

4.1. Routing Scheme

The routing algorithm is inspired by and based on [Cowen 2001; Thorup and Zwick 2001]. We also use a set of landmarks $A \subseteq V$, but different from [Cowen 2001; Thorup and Zwick 2001], we use $\mathsf{core}_{\gamma'}(G)$ as landmarks instead of nodes sampled at random. For each node u in G, let $\ell(u)$ denote u's closest landmark, that is, $\ell(u) := \arg\min_{v\in\mathsf{core}_{\gamma'}(G)} d(u,v)$. The local targets of node u are defined as the elements of its *ball* $B_G(u)$. Similar to the second scheme in [Thorup and Zwick 2001], each node u stores the ports to route messages along the shortest paths to all landmarks and to its local targets. If the target v is neither a landmark nor a local target of u, the message is routed to v's closest landmark $\ell(v)$ and from there to the target v.

The scheme is a labeled scheme. For a node u to know $\ell(v)$ of any target v, the address of node v contains an encoding of $\ell(v)$. Moreover, for a node w on the shortest path from $\ell(v)$ to v ($w \neq \ell(v)$ and $w \neq v$), v may not be in $B_G(w)$ and thus w may not know the port to route messages to v. To resolve this issue, we further extend the address of v by *encoding* the shortest path from the landmark $\ell(v)$ to v.

Let $(s = u_0, u_1, \ldots, u_m = t)$ denote the sequence of nodes on a shortest path from s to t. Let $SP(s,t)$ be the encoding of this shortest path as an array with m entries, wherein $SP(s,t)[i]$ denotes the port to route from u_i to u_{i+1} for all $i = 0, 1, \ldots, m-1$. Thus $SP(s,t)$ can be encoded with $\sum_{i=0}^{m-1} \log_2 \lceil \deg(u_i) \rceil$ bits. We now provide the precise definitions of addresses, message headers, and local routing tables.

Definition 4.1.

— The address of node $u \in V$ is $\texttt{addr}(u) := (u, \ell(u), SP(\ell(u), u))$.
— The header of a message from node s to node t is in one of the following formats:
  (1) $\texttt{header} = (route, s, t)$, where $route = \texttt{local}$,
  (2) $\texttt{header} = (route, s, addr)$, where $route = \texttt{toLandmark}$ and $addr = \texttt{addr}(t)$,
  (3) $\texttt{header} = (route, s, t, pos, SP)$, where $route \in \{\texttt{fromLandmark}, \texttt{direct}\}$, pos is a non-negative integer that may be modified along the route, and $SP = SP(s,t)$ if $route = \texttt{direct}$ or $SP = SP(\ell(t), t)$ if $route = \texttt{fromLandmark}$,
  (4) $\texttt{header} = (route, s, t, SP)$, where $route = \texttt{handshake}$ and SP is a reversed shortest path from t to s to be encoded along the path from s to t.
— The local routing table for each node u consists of the information about routes to the core and the information about local routes:

$$\texttt{tbl}(u) := \{(v, \texttt{port}_u(v)) : v \in \mathsf{core}_{\gamma'}(G)\} \cup \{(v, \texttt{port}_u(v)) : v \in B_G(u)\},$$

where $\texttt{port}_u(v)$ is the local port of u to route messages towards node v along some shortest path from u to v.

The routing procedure for the source is described in Algorithm 1, based on whether t is local or not and whether a shortest path to t is known due to an earlier handshake or not.

ALGORITHM 1: Sending a message using LANDMARKBALLROUTING, source s, target $t \neq s$.

if $t \in B_G(s)$ **then**
  send packet with $\texttt{header} = (\texttt{local}, s, t)$ using $\texttt{port}_s(t)$ stored in $\texttt{tbl}(s)$
else if s stored $SP(s,t)$ after a handshake **then**
  send packet with $\texttt{header} = (\texttt{direct}, s, t, 0, SP(s,t))$ using port $SP(s,t)[0]$
else
  send packet with $\texttt{header} = (\texttt{toLandmark}, s, \texttt{addr}(t))$ using $\texttt{port}_s(\ell(t))$ stored in $\texttt{tbl}(s)$
end if

The forwarding procedure is described in Algorithm 2, listing pseudocode for an intermediate node u to determine whether to forward the message using its local routing table (Lines 9 and 15), or to forward the message using the shortest path encoded in the header (Lines 11–13), or to switch the routing direction from towards the landmark $\ell(t)$ to towards the target t (Lines 5–7).

The correctness of the algorithm is based on the simple observation that if $t \in B_G(s) \cup \mathsf{core}_{\gamma'}(G)$ (and thus t is in the routing table of s), then, for all nodes w on the shortest path from s to t, we also have $t \in B_G(w) \cup \mathsf{core}_{\gamma'}(G)$.

An additional handshake protocol (Algorithm 3) handles the special case when $t \notin B_G(s)$ but $s \in B_G(t)$. In this case, the basic LANDMARKBALLROUTING scheme only

ALGORITHM 2: Forwarding a message at node u using LANDMARKBALLROUTING. Message from source s to target $t \neq s$ with header `header`.

1: **if** $u =$ `header`$.t$ **then**
2: **exit** as the packet arrived.
3: **end if**
4: **if** `header`$.route =$ `toLandmark` **then**
5: **if** $u =$ `header`$.addr.\ell(t)$ **then**
6: `header`$.route \leftarrow$ `fromLandmark`; `header`$.pos \leftarrow 0$; `header`$.SP \leftarrow$ `header`$.addr.SP(\ell(t), t)$;
7: forward packet with the new `header` using port `header`$.SP[0]$
8: **else**
9: forward the packet to $\texttt{port}_u(\texttt{header}.addr.\ell(t))$ stored in $\texttt{tbl}(u)$
10: **end if**
11: **else if** `header`$.route \in \{$`fromLandmark`, `direct`$\}$ **then**
12: `header`$.pos \leftarrow$ `header`$.pos + 1$
13: forward the packet using port `header`$.SP[$`header`$.pos]$
14: **else if** `header`$.route =$ `local` **then**
15: forward the packet using $\texttt{port}_u(\texttt{header}.t)$ stored in $\texttt{tbl}(u)$
16: **end if**

achieves worst-case stretch 5 instead of 3. However, t knows the reverse path from t to s. Since the graph is undirected, t can send a special `handshake` message back to s (Line 2), and each node along the path encodes the reverse port number such that, in the end, s knows the shortest path from s to t (Lines 3–10). For simplicity of exposition we use the reasonable assumption [Abraham et al. 2006b] that node u knows the port q on which the message is received. If this assumption does not hold, our handshake protocol can be adapted accordingly as follows. In the routing table of a node u, for all $v \in B_G(u) \cup \mathsf{core}_{\gamma'}(G)$, we also store a $\texttt{rev-port}_u(v) = \texttt{port}_w(u)$, where w is the first node on the path from u to v. Then, when forwarding the `handshake` message from t to s, every node u on the path (including t) prepends $\texttt{rev-port}_u(s)$ to the SP in the header. This increases the routing table size by at most $\lceil \log_2 n \rceil$ bits per entry. Note that, in Algorithm 3, we also include the case of $s \in \mathsf{core}_{\gamma'}(G)$ (see Line 1), in which case the stretch is improved from 3 to 1. The performance of Algorithms 2 and 3 is evaluated in the following theorem, which is proven in the next section.

ALGORITHM 3: Handshake protocol on node u upon the receipt of a packet from a port q with header `header`.

1: **if** `header`$.route =$ `fromLandmark` **and** $u =$ `header`$.t$ **and** `header`$.s \in B_G(u) \cup \mathsf{core}_{\gamma'}(G)$ **then**
2: send packet with `header` $= ($`handshake`$, u,$ `header`$.s, Nil)$ using $\texttt{port}_u(\texttt{header}.s)$ stored in $\texttt{tbl}(u)$.
3: **else if** `header`$.route =$ `handshake` **then**
4: `header`$.SP = q \cdot$ `header`$.SP$ /* prepend the port q as part of the reverse path */
5: **if** `header`$.t = u$ /* reach handshake destination */ **then**
6: store $SP(u,$ `header`$.s) =$ `header`$.SP$ locally for later use (see Line 3 of LANDMARKBALLROUTING.)
7: **else**
8: forward packet with the new `header` to $\texttt{port}_u(\texttt{header}.t)$ stored in $\texttt{tbl}(u)$.
9: **end if**
10: **end if**

THEOREM 4.2. LANDMARKBALLROUTING *together with the handshake protocol is a routing scheme with the following properties:*

— *the worst-case stretch is* 5 *without handshaking,*
— *the worst-case stretch is* 3 *after handshaking, and*
— *every routing decision takes constant time.*

Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ *be a constant. Assume Equation (1) is satisfied. For random graphs sampled from* $\mathbf{RPLG}(n, \tau)$ *with* n *sufficiently large,* LANDMARKBALLROUTING *has the following performance properties:*

— *the expected maximum table size is* $O(n^\gamma \log n)$ *bits (not taking into account additional information stored after handshaking); this bound also holds with probability at least* $1 - 1/n$*,*
— *address length and message header size are* $O(\log n \log \log n)$ *bits with probability* $1 - o(1)$*, and*
— *addresses and routing tables can be generated efficiently in expected time* $O(n^{1+\gamma} \log n)$ *and this bound also holds with probability at least* $1 - 1/n$*.*

5. ANALYSIS OF THE COMPACT ROUTING SCHEME

In this section, we analyze the performance of LANDMARKBALLROUTING for random power-law graphs.

5.1. Stretch

The proofs use the triangle inequality as in [Cowen 2001; Thorup and Zwick 2001].

LEMMA 5.1. LANDMARKBALLROUTING *has worst-case stretch* 5*. After handshaking with stretch* 5*,* LANDMARKBALLROUTING *has worst-case stretch* 3*.*

PROOF. By the triangle inequality [Cowen 2001], it is easy to verify the worst-case stretch 3 after handshaking. Before handshaking, the worse-case stretch happens when $t \notin B_G(s)$ and $s \in B_G(t)$. It holds that $d(s,t) \geq d(s, \ell(s))$. The radius of t's ball is at most $d(t, \ell(t)) \leq d(t, \ell(s)) \leq d(\ell(s), s) + d(s,t)$. Also, the distance from s to t's landmark is at most $d(s, \ell(t)) \leq d(s,t) + d(t, \ell(t))$. This results in a total path length of at most

$$d(s, \ell(t)) + d(\ell(t), t) \leq d(s,t) + 2d(t, \ell(t)) \leq d(s,t) + 2(d(\ell(s), s) + d(s,t)) \leq 5d(s,t).$$

□

5.2. Random Power-Law Graphs and their Cores and Balls

We first prove some properties of the adapted random power-law graph model. Let G be a random graph sampled from $\mathbf{RPLG}(n, \tau)$. For a set of nodes S, define its *volume* $Vol(S)$ as the sum of all its nodes' w_i, that is, $Vol(S) := \sum_{v_i \in S} w_i$. We abbreviate $Vol(G) = Vol(V(G))$. Note that $Vol(G) = 1/\rho$. Let $vol(S)$ denote the sum of the nodes' degrees in the actual graph G, $vol(S) := \sum_{v_i \in S} \deg_G(v_i)$. The following lemma proves that $Vol(G)$ is linear in n.

LEMMA 5.2. *Let* G *be a random graph sampled from* $\mathbf{RPLG}(n, \tau)$ *with* n *sufficiently large. The volume* $Vol(G)$ *satisfies*

$$n < Vol(G) \leq \frac{\tau - 1}{\tau - 2} n.$$

PROOF. Lower bound: it holds that $\sum_k w_k > n$, as $\forall k < n : w_k > 1$ and $w_n = 1$. Upper bound: it holds that

$$Vol(G) = \sum_{k=1}^{n} w_k < w_1 + \int_1^n \left(\frac{n}{x}\right)^{1/(\tau-1)} \mathrm{d}x \leq \frac{\tau - 1}{\tau - 2} n.$$

□

In the following, we show concentration results for the actual degree of a vertex and for the volume of a set of vertices in the adapted $\mathbf{RPLG}(n,\tau)$ model. We also restate the corresponding results in the original **FDRG** model. The basic idea to prove the results for the $\mathbf{RPLG}(n,\tau)$ model is to split the random variable for the degree D_i of node v_i into deterministic and random edges and then bound both parts individually.

LEMMA 5.3 ([CHUNG AND LU 2006, LEMMA 5.6][MCDIARMID 1998, THEOREM 2.7]). *For a random graph sampled from $\mathbf{FDRG}(\vec{w})$, the random variable D_i measuring the degree of vertex v_i is concentrated around its expectation w_i as follows:*

$$\Pr[D_i > w_i - c\sqrt{w_i}] \;\geq\; 1 - e^{-c^2/2} \tag{2}$$

$$\Pr[D_i < w_i + c\sqrt{w_i}] \;\geq\; 1 - e^{-\frac{c^2}{2(1+c/(3\sqrt{w_i}))}} \tag{3}$$

LEMMA 5.4 ([CHUNG AND LU 2006, LEMMA 5.9]). *For a random graph sampled from $\mathbf{FDRG}(\vec{w})$, for a subset of vertices S and for all $0 < c \leq \sqrt{Vol(S)}$,*

$$\Pr[|vol(S) - Vol(S)| < c\sqrt{Vol(S)}] \geq 1 - 2e^{-c^2/6}.$$

LEMMA 5.5. *For a random graph sampled from $\mathbf{RPLG}(n,\tau)$ with n sufficiently large, if $w_i \geq 32\ln n$, for vertex v_i, the degree D_i satisfies the following:* $\Pr[w_i/4 \leq D_i \leq 3w_i] > 1 - 2/n^4$.

PROOF. Recall that $\rho = 1/Vol(G) < 1/n$ (by Lemma 5.2). For $1 \leq i \leq n$, let $h(i) \in [1,n]$ denote the smallest integer such that $\rho w_{h(i)} w_i \leq 1$. Consider $h(1)$. Since $\rho w_1 (\frac{n}{n^{3-\tau}})^{1/(\tau-1)} \leq 1$, we have that $h(1) \leq \lceil n^{3-\tau} \rceil$. Therefore, for all $1 \leq i \leq n$, $h(i) \leq h(1) \leq \lceil n^{3-\tau} \rceil$.

We split the degree D_i into two parts: the contribution by edges to nodes v_j with $j < h(i)$ and the contribution stemming from edges to nodes v_j with $j \geq h(i)$. When $h(i) \geq 1$, there are at least $h(i) - 1$ edges to nodes v_j with $j \leq h(i)$. Now consider the edges between v_i and v_j for $j \geq h(i)$. Since the sequence $\vec{w}$ is monotonically decreasing, and since $n^{3-\tau} \geq 1$ and $n \geq 4^{\frac{\tau-1}{(\tau-2)^2}}$,

$$\begin{aligned} \sum_{j=h(i)}^{n} w_j \;&\geq\; \int_{n^{3-\tau}+1}^{n} (n/x)^{1/(\tau-1)} \mathrm{d}x \\ &\geq\; \frac{\tau-1}{\tau-2}\left(n - n^{1/(\tau-1)} 2^{\frac{\tau-2}{\tau-1}} n^{\frac{\tau-2}{\tau-1}(3-\tau)}\right) \\ &\geq\; \frac{\tau-1}{2(\tau-2)} n. \end{aligned}$$

Recall that $\rho = 1/\sum_{i=1}^{n} w_i \geq \frac{\tau-2}{n(\tau-1)}$ by Lemma 5.2. Let D_i' be the random variable denoting the number of edges from v_i to v_j with $j \geq h(i)$ in a random graph. Thus, $E[D_i'] = \mu = \rho w_i \sum_{j=h(i)}^{n} w_i \geq w_i/2 \geq 16\ln n$. Also $\mu \leq w_i$. Since there are no deterministic edges in this case, the random variable D_i' can be bounded using Lemma 5.3:

$$\begin{aligned} \Pr[D_i' > \mu/2] \;&\geq\; 1 - e^{-\mu/4} \geq 1 - 1/n^4, \\ \Pr[D_i' < 2\mu] \;&\geq\; 1 - e^{-3\mu/8} \geq 1 - 1/n^4. \end{aligned}$$

If $h(i) = 1$, the lemma follows directly. If $h(i) > 1$, we have $D_i \leq D_i' + h(i) - 1$. Notice that $\rho w_i (n/w_i)^{1/(\tau-1)} \leq 1$, which implies that $h(i) \leq \lceil w_i \rceil \leq w_i + 1$. Therefore,

$$\Pr[w_i/4 \leq \mu/2 \leq D_i \leq 3w_i] \leq 1 - 2/n^4.$$

□

LEMMA 5.6. *Let G be a random graph sampled from* $\mathbf{RPLG}(n, \tau)$ *with n sufficiently large. For a subset of vertices S satisfying $Vol(S) \geq 192 \ln n$, it holds with probability at least $1 - 2/n^3$ that $Vol(S)/8 \leq vol(S) \leq 4\,Vol(S)$.*

PROOF. We split S into two parts. Nodes v_i with small w_i, $S_1 := \{v_i \in S : w_i < 32 \ln n\}$, and nodes v_i with large w_i, $S_2 = S \backslash S_1$. By Lemma 5.5, $\Pr[\,Vol(S_2)/4 \leq vol(S_2) \leq 3\,Vol(S_2)] \geq 1 - 2\,|S_2|\,/n^4$.

As for each vertex $v_i \in S_1$, $w_i < 32 \ln n$, we can apply Lemma 5.4 to S_1, since no deterministic edges are attached to S_1. Therefore, if $Vol(S_1) \geq 96 \ln n$, by Lemma 5.4, $\Pr[\,Vol(S_1)/2 \leq vol(S_1) \leq 2\,Vol(S_1)/3] \geq 1 - 2/n^4$. Therefore, the statement holds with probability at least $1 - 2(|S_2| + 1)/n^4 \geq 1 - 2/n^3$.

If $Vol(S_1) < 96 \ln n$, we have $Vol(S_2) \geq Vol(S)/2 \geq 96 \ln n$. Nevertheless, we can still apply Lemma 5.4 to bound $vol(S_1)$ from above as $\Pr[vol(S_1) < \frac{3}{2} \cdot 96 \ln n \leq \frac{3}{4}\,Vol(S)] \geq 1 - 2/n^4$. In this case, since $\Pr[\,Vol(S)/8 \leq Vol(S_2)/4 \leq vol(S_2) \leq 3\,Vol(S_2)] \geq 1 - 2\,|S_2|\,/n^4$, the statement also holds with probability at least $1 - 2/n^3$. □

COROLLARY 5.7. *The number of edges of a random graph sampled from* $\mathbf{RPLG}(n, \tau)$ *with n sufficiently large is at most $vol(G)/2 \leq \frac{4(\tau-1)}{\tau-2} n$ with probability at least $1 - 1/n^2$.*

There is an edge between two nodes v_i, v_j with probability proportional to w_i and w_j. This is generalized for sets of nodes $S, T \subseteq V(G)$ in the following and holds for both $\mathbf{FDRG}(\vec{w})$ and $\mathbf{RPLG}(n, \tau)$.

LEMMA 5.8 ([CHUNG AND LU 2002, LEMMA 3.3], PROOF IN [LU 2002B, LEMMA 9]). *For any two disjoint subsets S and T with $Vol(S) \cdot Vol(T) > c \cdot Vol(G)$, we have*

$$\Pr[d(S,T) > 1] = \prod_{v_i \in S, v_j \in T} \max\{0, (1 - w_i w_j / Vol(G))\} \leq e^{-Vol(S) \cdot Vol(T)/Vol(G)} \leq e^{-c}.$$

5.3. Core Size

To compute the size of $\mathsf{core}_{\gamma'}(\vec{w})$, we solve the inequality $w_k > n^{\gamma'}$ and obtain k.

$$\begin{aligned} w_k \;&=\; \left(\frac{n}{k}\right)^{\frac{1}{\tau-1}} > n^{\gamma'} \Leftrightarrow k^{-\frac{1}{\tau-1}} > n^{\gamma' - \frac{1}{\tau-1}} \\ &\Leftrightarrow\; k < n^{(1-\tau)(\gamma' - \frac{1}{\tau-1})} = n^{\gamma'(1-\tau)+1} \end{aligned}$$

As $\gamma' = \frac{1-\gamma}{\tau-1}$, we have $|\mathsf{core}_{\gamma'}(\vec{w})| = \lceil n^{\gamma'(1-\tau)+1} \rceil - 1 = \lceil n^{\gamma} \rceil - 1$.

Even if the same degree threshold $n^{\gamma'}$ is used for $\mathsf{core}_{\gamma'}(\vec{w})$ and $\mathsf{core}_{\gamma'}(G)$, the two sets of nodes may differ. For a slightly smaller degree threshold $n^{\gamma'}/4$ (as in Definition 3.3), the core of the actual graph contains $\mathsf{core}_{\gamma'}(\vec{w})$ with high probability (apply Lemma 5.5).

LEMMA 5.9. *Let G be a random graph sampled from* $\mathbf{RPLG}(n, \tau)$ *with n sufficiently large. With probability at least $1 - 1/n^2$ it holds that $\mathsf{core}_{\gamma'}(\vec{w}) = \{v_i : w_i > n^{\gamma'}\} \subseteq \{v_i : \deg(v_i) > n^{\gamma'}/4\} = \mathsf{core}_{\gamma'}(G)$.*

PROOF. Let v_i be a vertex in $\mathsf{core}_{\gamma'}(\vec{w})$. By Lemma 5.5, $D_i \geq n^{\gamma'}/4$ with probability at least $1 - 2/n^4$. This holds for all $j \leq i$. Therefore, by union bound, the probability that $\mathsf{core}_{\gamma'}(\vec{w}) \subseteq \{v_i : \deg(v_i) > n^{\gamma'}/4\}$ is at least $1 - 1/n^2$. □

LEMMA 5.10. *Let G be a random graph sampled from* $\mathbf{RPLG}(n, \tau)$ *with n sufficiently large. With probability at least $1 - 1/n^2$, $|\mathsf{core}_{\gamma'}(G)| = \Theta(n^{\gamma})$.*

PROOF. Since $\mathsf{core}_{\gamma'}(G)$ contains $\mathsf{core}_{\gamma'}(\vec{w})$ with high probability ($1-1/n^2$), its size is at least n^γ with high probability.

Let $i = 144n^\gamma$. By Lemma 5.5, $D_i \le 3w_i < n^{\gamma'}/4$ with probability at least $1-2/n^4$. This holds for all $j \in (i, n]$. By union bound, $\mathsf{core}_{\gamma'}(G)$ does not contain any vertex v_j for $i \le j \le n$, with probability at least $1-1/n^2$, which implies $|\mathsf{core}_{\gamma'}(G)| \le 144n^\gamma$ with probability at least $1-1/n^2$. □

5.4. Ball Sizes

Let G be a random graph sampled from random power-law graph. Recall that a ball is defined by

$$B_G(u) = \{v \in V(G) : d(u,v) < \min_{v' \in \mathsf{core}_{\gamma'}(G)} d(u,v')\}.$$

LEMMA 5.11. *Let $\beta = \gamma'(\tau-2) + \frac{(2\tau-3)\varepsilon}{\tau-1}$ be a constant. Assume Equation (1) is satisfied. For a random graph G sampled from* $\mathbf{RPLG}(n,\tau)$*, with probability at least $1-3/n^2$, it holds that for all $u \in V(G)$,*

$$\begin{aligned} |B_G(u)| &= |\{u' \in V(G) : d(u,u') < d(u, \mathsf{core}_{\gamma'}(\vec{w}))\}| = O(n^\beta), \\ |E(B_G(u))| &= O(n^\beta \log n), \end{aligned}$$

where $E(B_G(u))$ is the set of internal edges among vertices in $B_G(u)$.

Since for $\mathbf{RPLG}(n,\tau)$ the edges are independent, in our analysis, the existence of every edge in random graph G is only determined when it is needed, and before that it is treated as a probability distribution as defined in our random graph model. We call the determination of the existence of an edge according to its probability distribution *revealing* the edge.

For a given vertex $u \in V(G)$, we define a sequence of balls $(B_0 = \{u\}, B_1, B_2, \ldots)$ as follows: Let $V' = V \setminus \mathsf{core}_{\gamma'}(\vec{w})$. Now define $B_0 = \{u\}$ and $B_i = \{v : d_G(u,v) \le i\}$. We also define the circles $C_i = B_i \setminus B_{i-1}$ for $i \ge 0$ with $B_{-1} = \emptyset$. Let E_i be the number of edges between C_i and $C_i \cup C_{i+1}$. We first give a concentration result on E_i.

LEMMA 5.12. *For circle C_i, the following holds with probability at least $1-2/n^3$:*

$$\begin{aligned} &\textit{If } Vol(C_i) < 192 \ln n, \textit{ then } E_i \le 4 \cdot 192 \ln n, and \\ &\textit{if } Vol(C_i) \ge 192 \ln n, \textit{ then } E_i \le 4\,Vol(C_i). \end{aligned}$$

PROOF. For our analysis, we assume that the edges of the random graph are revealed in consecutive steps as follows: in step i with $i \ge 0$, edges from C_i to $V' \setminus B_{i-1}$ are revealed and circle C_{i+1} is formed. In other words, when discovering C_i, the edges between C_i and $V'' = V' \setminus B_{i-1}$ have not been revealed yet.

In particular, E_i measures the number of edges between C_i and V'' under the condition that we know all edges adjacent to B_{i-1}. We can define another random graph G' on the vertex set V'', such that the edge between two vertices in V'' is sampled with the same probability as in $\mathbf{RPLG}(n,\tau)$. Clearly, E_i and $vol_{G'}(C_i)$ have the same distribution, where $vol_{G'}(C_i)$ denotes the number of edges adjacent to C_i in G'.

Let $vol(C_i)$ denote the random variable measuring the number of edges adjacent to C_i in the original model **FDRG**. $vol_{G'}(C_i)$ is *stochastically dominated* by $vol(C_i)$. Hence, the lemma directly follows since it applies to $vol(C_i)$ by Lemma 5.6. □

Since there are at most n circles, Lemma 5.12 holds for all circles with probability at least $1-2/n^2$. We are now ready to prove Lemma 5.11.

PROOF OF LEMMA 5.11. Let k be the smallest integer such that $Vol(B_k) \ge n^\beta$. We have the conditions $Vol(B_k) \ge n^\beta$, $Vol(\mathsf{core}_{\gamma'}(\vec{w})) \ge |\mathsf{core}_{\gamma'}(\vec{w})|\, n^{\gamma'} = n^{\gamma+\gamma'}$, and

$Vol(G) \leq \frac{\tau-1}{\tau-2}n$ (Lemma 5.2). From Equation (1), $n^{\beta-\gamma'(\tau-2)} > 2\frac{\tau-1}{\tau-2}\ln n$. Since the edges between B_k and $\mathsf{core}_{\gamma'}(\vec{w})$ have not been revealed, Lemma 5.8 can be applied. Due to Lemma 5.8, there is an edge between B_k and $\mathsf{core}_{\gamma'}(\vec{w})$ with probability at least $1-1/n^2$.

$$\begin{aligned}
\beta &= \gamma'(\tau-2) + \frac{(2\tau-3)\varepsilon}{\tau-1} \\
\gamma' &= \frac{1-\gamma}{\tau-1} \\
\gamma &= \frac{\tau-2}{2\tau-3} + \varepsilon \\
Vol(B_k) &\geq n^\beta \\
Vol(\mathsf{core}_{\gamma'}(\vec{w})) &\geq |\mathsf{core}_{\gamma'}(\vec{w})|\, n^{\gamma'} = n^{\gamma+\gamma'} \\
Vol(B_k) \cdot Vol(\mathsf{core}_{\gamma'}(\vec{w})) &\geq n^\beta \cdot n^{\gamma+\gamma'} \\
&= n^{\gamma'(\tau-2)+\frac{(2\tau-3)\varepsilon}{\tau-1}+\gamma+\gamma'} \\
&= n^{\gamma'(\tau-1)+\frac{(2\tau-3)\varepsilon}{\tau-1}+\gamma} \\
&= n^{1+\frac{(2\tau-3)\varepsilon}{\tau-1}} \qquad \text{next, apply Equation (1)} \\
&\geq n \cdot \frac{2(\tau-1)}{\tau-2} 2\ln n \qquad \text{next, apply Lemma 5.2} \\
&\geq Vol(G) \cdot 2\ln n.
\end{aligned}$$

Recall that $\mathsf{core}_{\gamma'}(\vec{w}) \subseteq \mathsf{core}_{\gamma'}(G)$ with probability at least $1-1/n^2$ by Lemma 5.9. Hence $B_G(u) \subseteq B_k$ with probability at least $1-2/n^2$.

In the following, we bound the size of B_k. Lemma 5.12 holds for all circles with high probability. In our case, $Vol(C_{k-1}) \leq Vol(B_{k-1}) < n^\beta$. By Lemma 5.12, $|C_k| \leq E_{k-1} \leq 4n^\beta$ with probability at least $1-1/n^2$. Then, $|B_k| = |B_{k-1}| + |C_k| \leq Vol(B_{k-1}) + |C_k| \leq 5n^\beta$.

Since $B_G(u) \subseteq B_k$ with probability at least $1-2/n^2$, we have $|E(B_G(u))| = O(vol(B_{k-1}(u))) = O\left(\sum_{i=0}^{k-1} E_i\right)$, with probability at least $1-2/n^2$.

By Lemma 5.12, with probability at least $1-1/n^2$, $E_i \leq 4 \cdot 192 \ln n + 4\,Vol(C_i)$ for all i. Since $k \leq n^\beta$, with probability at least $1-3/n^2$,

$$|E(B_G(u)| = O\left(\sum_{i=0}^{k-1} E_i\right) = O(4 \cdot 192 n^\beta \ln n + 4\,Vol(B_{k-1})) = O(n^\beta \log n).$$

□

5.5. Table Sizes and Computations

The core $\mathsf{core}_{\gamma'}(G)$ has size $\Theta(n^\gamma)$ with probability at least $1-1/n^2$ (Lemma 5.10) and all balls $B_G(u)$ have size $O(n^\beta)$ with probability at least $1-3/n^2$ (Lemma 5.11). Therefore, we have the following result.

LEMMA 5.13. *For a random graph G sampled from* **RPLG**(n,τ)*, for all $u \in V(G)$, the expected table size is at most*

$$|\mathtt{tbl}(u)| = O(n^\gamma)$$

and all tables can be generated in expected time at most $O(n^{1+\gamma}\log n)$. These bounds also hold with probability at least $1-1/n$.

PROOF. Note that each entry of $\texttt{tbl}(u)$ has $O(\log n)$ bits. Thus the total table size per node is $O(n^\gamma \log n)$ bits.

Our algorithm is deterministic. The expected time (space) complexity is the average running time (space) of our algorithm over all graphs from the random graph distribution $\mathbf{RPLG}(n, \tau)$.

Given a graph G with n nodes and m edges, our algorithm computes the core $\mathsf{core}_{\gamma'}(G)$ of G with time complexity $O(m + n \log n)$. It runs a complete breadth-first search for each node of the core in time $O(m)$. Let $B_G(u)$ be the ball computed in our algorithm for vertex u. Let $T(B_G(u))$ denote the time to compute $B_G(u)$. Therefore, the time complexity TC and space complexity SC of our algorithm are at most

$$TC(G) \;=\; O\left(m \cdot |\mathsf{core}_{\gamma'}(G)| + \sum_{v \in V(G)} T(B_G(u))\right), \tag{4}$$

$$SC(G) \;=\; O\left(n \cdot |\mathsf{core}_{\gamma'}(G)| + \sum_{v \in V(G)} |B_G(u)|\right). \tag{5}$$

We now know that with probability at least $1 - 5/n^2$, all of the following conditions are true: (1) $m = \Theta(n)$ (Corollary 5.7); (2) $|\mathsf{core}_{\gamma'}(G)| = \Theta(n^\gamma)$ (Lemma 5.10); (3) $|B_G(u)| = O(n^\beta)$ for all vertices u (Lemma 5.11); (4) $T(B_G(u)) = O(n^\beta \log n)$ for all vertices u (Lemma 5.11). Therefore, from Equations (4) and (5), we know that with probability at least $1 - 5/n^2$, the space complexity of our algorithm is $O(n^{1+\gamma} + n^{1+\beta})$ and the time complexity is $O(n^{1+\gamma} + n^{1+\beta} \log n)$.

Finally, we fix the parameters to obtain a balanced scheme. In a balanced scheme, the core size and the expected ball sizes are asymptotically equivalent, that is, $\beta = \gamma$.

$$\begin{aligned} \beta &= \gamma'(\tau - 2) + \frac{(2\tau - 3)\varepsilon}{\tau - 1} \text{ and} \\ \gamma' &= \frac{1 - \gamma}{\tau - 1}, \text{ we have} \\ \gamma &= \frac{\tau - 2}{2\tau - 3} + \varepsilon \end{aligned}$$

Therefore, assuming that Equation (1) is satisfied, the space requirement per node is $O(n^\gamma \log n)$ bits and the preprocessing time is bounded by $O(n^{1+\gamma} \log n)$, which holds with probability at least $1 - 1/n$. □

5.6. Address Lengths

We now bound the number of bits for the address of each vertex. For one vertex u, its address contains the encoding of the shortest path $SP(u, \ell(u))$ from u to its landmark $\ell(u)$. We need to bound the diameter of a random power-law graph and the diameter of its core.

LEMMA 5.14 (CHUNG AND LU [CHUNG AND LU 2002, CLAIM 4.4]). *For a random graph sampled from $\mathbf{RPLG}(n, \tau)$, with probability at least $1 - o(1)$, the diameter of its largest connected component is $O(\log n)$.*

By Lemma 5.14, the length of $SP(u, \ell(u))$ is at most $O(\log n)$ asymptotically almost surely. Therefore, $SP(s, t)$ can be encoded with $O(\log^2 n)$ bits. This bound can be improved to $O(\log n \cdot \log \log n)$, as proven in the following lemma.

LEMMA 5.15. *For a random graph G sampled from $\mathbf{RPLG}(n, \tau)$, with probability at least $1 - o(1)$, it holds that for all $s, t \in V(G)$, $SP(s, t)$ can be encoded with $O(\log n \log\log n)$ bits.*

The proof is split into several claims from [Chung and Lu 2002]. We first extend the core.

Definition 5.16. The *extended core* of a random graph from $\mathbf{RPLG}(n, \tau)$ contains all nodes v_i with w_i at least $n^{1/\log\log n}$, that is, $\mathsf{core}^+(\vec{w}) = \{v_i \in V : w_i \geq n^{1/\log\log n}\}$.

Note that, as τ is a constant, $1/\log\log n \leq \gamma'$ for large enough n, and thus $\mathsf{core}^+(\vec{w}) \supseteq \mathsf{core}_{\gamma'}(\vec{w})$. The following lemma constitutes a bound for the diameter of the core.

LEMMA 5.17 (CHUNG AND LU [CHUNG AND LU 2002, CLAIM 4.1]). *Let G be a random graph sampled from $\mathbf{RPLG}(n, \tau)$ with n sufficiently large. The diameter of the subgraph induced by $\mathsf{core}^+(\vec{w})$ in G is $O(\log\log n)$ with probability at least $1 - 1/n$.*

The next lemma states that a vertex v_i with large enough w_i is close to the *extended core*.

LEMMA 5.18 (CHUNG AND LU [CHUNG AND LU 2002, CLAIM 4.2]). *Let G be a random graph sampled from $\mathbf{RPLG}(n, \tau)$. There exists a constant C, such that each vertex v_i with $w_i \geq \log^C n$ is at distance $O(\log\log n)$ from the* extended core*, with probability at least $1 - 1/n^2$.*

COROLLARY 5.19 (COROLLARY OF LEMMA 5.18). *Let G be a random graph sampled from $\mathbf{RPLG}(n, \tau)$ with n sufficiently large. Let C be the constant in Lemma 5.18. With probability at least $1 - 1/n$, the distance between any two vertices v_i, v_j with $w_i \geq \log^C n$ and $w_j \geq \log^C n$ is $O(\log\log n)$.*

The power-law degree sequence of our $\mathbf{RPLG}(n, \tau)$ with $\tau \in (2, 3)$ is chosen such that the magnitude of each element w_i is at least as large as the corresponding degree potential in [Chung and Lu 2002]. The sequence of edge probabilities in [Chung and Lu 2002] is stochastically dominated by the sequence of $\mathbf{RPLG}(n, \tau)$. Sampling a graph from each of the distributions simultaneously, we obtain that, intuitively, a graph sampled from $\mathbf{RPLG}(n, \tau)$ "contains a subgraph" from **FDRG**. Distances in a graph sampled from $\mathbf{RPLG}(n, \tau)$ can only be shorter than in the corresponding graph sampled from **FDRG**. Consequently, Lemma 5.17 and Lemma 5.18 can be directly extended to $\mathbf{RPLG}(n, \tau)$. The proof of Lemma 5.14 in [Chung and Lu 2002] is based on the volume expansion. Since our graph has larger w_i values, the volume expansion arguments can be directly applied and Lemma 5.14 holds for our graph as well. We are now ready to prove Lemma 5.15.

PROOF OF LEMMA 5.15. Let v_i and v_j be the first and the last vertex in $SP(s, t)$ from s to t such that w_i and w_j both are greater than $\log^C n$, where C is the constant from Lemma 5.18. By Corollary 5.19, with probability $1 - 1/n$, the portion of the shortest path $SP(s, t)$ between v_i and v_j has length at most $O(\log\log n)$. Therefore, this portion of the shortest path can be encoded with $O(\log n \log\log n)$ bits, with probability $1 - 1/n$.

For the rest of the shortest path, each node has w_i at most $\log^C n$. By Lemma 5.5, all such nodes have degree at most $3\log^C n$ with probability at least $1 - 2/n^3$. To encode the next neighbor in the shortest path, at most $O(\log\log n)$ bits are necessary. Since $SP(s, t)$ contains $O(\log n)$ nodes with probability $1 - o(1)$ (Lemma 5.14), the rest of the shortest path can also be encoded with $O(\log n \log\log n)$ bits, with probability $1 - o(1)$. □

COROLLARY 5.20. *For a random graph G sampled from $\mathbf{RPLG}(n, \tau)$ with n sufficiently large, with probability at least $1 - o(1)$, it holds that for all $u \in V(G)$, the address* $\texttt{addr}(u)$ *can be encoded with $O(\log n \log \log n)$ bits.*

5.7. Additional Remarks

We give a more detailed explanation on why our analysis does not work to bound cluster sizes.

Our analysis crucially relies on Lemma 5.8 (from [Chung and Lu 2002]), using which we bound the ball sizes $|B(u)|$. When trying to bound the size of the cluster $C(u)$, the analysis breaks as follows: the lemma can be applied only in random graphs in which the edges between two sets $C(u)$ and A are independently and randomly selected according to the random power-law graph model, *after* the sets $C(u)$ and A have been fixed. In other words, to apply Lemma 5.8, one has to first fix two sets $C(u)$ and A *without* revealing any random edges between $C(u)$ and A in the random graphs. After set A is determined, $C(u)$ is defined to be the set $\{v : d(v, u) < d(v, A)\}$. But to determine $C(u)$, one must know the distance from v to A, which means the random edges between v and A have been revealed, and in particular, if there is an edge between v and some node in A, then v is not in $C(u)$ (since for all $v \in C(u)$, we have that $d(v, A) > d(v, u) \geq 1$). In this case, $C(u)$ has been biased to be a set "far away" from A, and since the randomness of the edges between $C(u)$ and A is no longer there, one can no longer apply Lemma 1 to derive an upper bound on the size of $C(u)$.

On the contrary, when we use balls, those balls $B_0, B_1, \ldots$ can be determined without revealing any edges between the balls and the core. Therefore we can still apply Lemma 5.8 to limit the sizes of the balls.

6. EXPERIMENTAL EVALUATION OF THE COMPACT ROUTING SCHEME

In this section, we experimentally demonstrate the efficiency of our scheme. We use the following datasets in our experiments.

Real-world graphs.. The most important application scenario for a compact routing scheme is arguably a communication network. The router-level topology of a portion of the Internet,[2] measured by CAIDA [CAIDA Cooperative Association for Internet Data Analysis 2003], is an undirected, unweighted graph with 190,914 nodes and 607,610 edges. The estimated power-law exponent (maximum likelihood method [Newman 2005]) is $\hat{\tau} = 2.82$.

Random power-law graphs.. We extracted the largest connected component from the random power-law graphs generated by Brady and Cowen [2006] (pre-generated graphs, N =10,000 and $\tau \in (2, 3)$, downloaded from `http://digg.cs.tufts.edu/`). In the results, we call them *BC Graphs*.

In addition, we generated graphs on 10,000 nodes with the tool BRITE [Medina et al. 2001] using the configurations for the Barabási [Barabási and Albert 1999] and Waxman [Waxman 1988] models for an Autonomous System Topology (AS) and a Router Topology (RT) — the precise configurations are listed in Section A. The number of edges generated is roughly 20,000. Edge weights were ignored and the links were interpreted as undirected.

Note that for all the random graphs considered, the generation process does not exactly match the $\mathbf{RPLG}(n, \tau)$.

[2]It is unclear whether the measurements provide an accurate representation of the Internet [Roughan et al. 2011].

Graph	CAIDA	ASBarabasi	RTBarabasi	ASWaxman	RTWaxman
random, $p = n^{-1/2}$	929.84±95.40	204.03±25.57	208.32±22.21	221.95± 24.73	217.75± 28.00
highdeg, $\lceil n^{\gamma} \rceil$	173.68±55.80	32.16±41.30	44.95±58.21	139.45±142.94	130.65±131.78
BC Graphs	$\tau = 2.1$	$\tau = 2.2$	$\tau = 2.3$	$\tau = 2.4$	$\tau = 2.5$
random, $p = n^{-1/2}$	74.90±37.96	74.94±44.78	77.49±50.56	79.74± 55.50	82.54± 60.17
highdeg, $\lceil n^{\gamma} \rceil$	55.20±67.48	48.50±54.57	42.20±42.94	43.28± 40.10	43.55± 38.37
BC Graphs	$\tau = 2.6$	$\tau = 2.7$	$\tau = 2.8$	$\tau = 2.9$	
random, $p = n^{-1/2}$	86.88±69.69	85.56±71.35	84.69±73.87	76.65± 71.71	
highdeg, $\lceil n^{\gamma} \rceil$	45.59±39.59	50.24±46.08	56.48±56.26	46.85± 46.65	

Table I. Table sizes: mean and standard deviation

Routing schemes.. In the specification of our routing scheme LANDMARKBALLROUTING, we use $n^{\gamma'}/4$ as a degree threshold (Definition 3.3) and obtain a core of size $\Theta(n^{\gamma})$. The largest connected components of the graphs generated by Brady and Cowen [2006] and the graphs generated using BRITE [Medina et al. 2001] do not contain nodes with such a high degree. Therefore, for the experiments with our routing scheme, the algorithm selects the $\lceil n^{\gamma} \rceil$ nodes with the highest degrees as landmarks. In practice, this might indeed be a better strategy.

We compare our high-degree selection strategy with the random selection with probability $n^{-1/2}$, which is *similar* to Thorup and Zwick [2001] for $k = 2$. Recall that their scheme is not optimized for power-law graphs but works for general, weighted graphs as well. We also compare our scheme with the values obtained by Brady and Cowen [2006].

Settings and results.. For the graphs generated by Brady and Cowen [2006], the high-degree selection and the random sampling process were executed five times for each of the ten graphs per value of τ, which gives a total of $5 \cdot 10 \cdot 9 \cdot 2 = 900$ routing scheme constructions. For each of the remaining graphs (Barabási, Waxman, CAIDA), both schemes were constructed at least 10 times. We report the table sizes (mean and standard deviation) in Table I. For each instance, 200 random (s, t) pairs were generated and packets routed. The multiplicative stretch (the length of the route divided by the length of a shortest path) is reported in Table II. The additive stretch (the length of the route *minus* the length of a shortest path) is reported in Table III.

In our experiments, the strategy of selecting few high-degree nodes as landmarks always produces significantly smaller routing tables compared to a large number of landmarks selected at random. The best results are achieved for the graphs stemming from the Barabási model, for which the high-degree-based tables are roughly 5 times smaller than their random-based counterpart. The average table size for the randomly selected landmarks is close to $\sqrt{n}$, which means that most balls are actually (almost) empty. As predicted by our analysis, this indicates that, for power-law graphs, the optimal balance for randomly selected landmarks may be smaller than $O(\sqrt{n})$.

Graph	CAIDA	ASBarabasi	RTBarabasi	ASWaxman	RTWaxman
random	1.28 ±0.16	1.38 ±0.28	1.38 ±0.25	1.37 ±0.25	1.38 ±0.16
highdeg, $\lceil n^\gamma \rceil$	1.12 ±0.14	1.15 ±0.21	1.20 ±0.22	1.36 ±0.26	1.35 ±0.24
BC Graphs	$\tau = 2.1$	$\tau = 2.2$	$\tau = 2.3$	$\tau = 2.4$	$\tau = 2.5$
random, $p = n^{-1/2}$	1.340±0.240	1.350±0.243	1.347±0.251	1.342±0.259	1.335±0.261
highdeg, $\lceil n^\gamma \rceil$	1.300±0.239	1.264±0.230	1.226±0.227	1.211±0.226	1.183±0.221
BC Graphs	$\tau = 2.6$	$\tau = 2.7$	$\tau = 2.8$	$\tau = 2.9$	
random, $p = n^{-1/2}$	1.330±0.275	1.306±0.281	1.290±0.286	1.247±0.284	
highdeg, $\lceil n^\gamma \rceil$	1.160±0.218	1.151±0.222	1.147±0.237	1.111±0.216	

Table II. Multiplicative stretch: mean and standard deviation

BC Graphs	$\tau = 2.1$	$\tau = 2.2$	$\tau = 2.3$	$\tau = 2.4$	$\tau = 2.5$
random, $p = n^{-1/2}$	1.744±1.001	1.955±1.098	2.112±1.243	2.302±1.405	2.512±1.617
highdeg, $\lceil n^\gamma \rceil$	1.532±1.015	1.463±1.069	1.360±1.158	1.394±1.266	1.342±1.373
BC Graphs	$\tau = 2.6$	$\tau = 2.7$	$\tau = 2.8$	$\tau = 2.9$	
random, $p = n^{-1/2}$	2.752±1.903	2.946±2.254	3.250±2.736	3.173±3.124	
highdeg, $\lceil n^\gamma \rceil$	1.309±1.514	1.442±1.824	1.608±2.266	1.384±2.331	

Table III. Additive stretch: mean and standard deviation

The average stretch is surprisingly consistent among different datasets. Even though there are fewer landmarks, the average stretch is better if high-degree nodes are selected as landmarks. Brady and Cowen [2006] claim average stretch 1.18–1.25 for the scheme by Thorup and Zwick [2001]. Our experiments do not confirm this claim: randomly selected nodes (similar to TZ) did not achieve this stretch. A possible reason for this could be that we use balls instead of clusters for short-range distances. Brady and Cowen also claim average stretch 1.11–1.22 for their scheme and small values for $\tau \in \{2.1, 2.2, 2.3\}$. Our scheme, except for the graphs of the Waxman model and for small values of $\tau \leq 2.2$, also achieves these average stretch values. The worst-case stretch is difficult to compare as our scheme has a (non-experimental) worst-case *multiplicative* stretch and the scheme by Brady and Cowen has an experimental worst-case *additive* stretch. Brady and Cowen conclude from their topology experiments that, for graphs up to 40,000 nodes, their scheme has a worst-case additive stretch of 10 while maintaining $O(\log^2 n)$-bit tables per node. For nodes 'close' to each other (distance less than 5), the multiplicative stretch of 3 yields better stretch guarantees. For nodes 'far' from each other (distance at least 5), the additive stretch of 10 yields better stretch guarantees. In power-law graphs, most distances are short, the typical distance being $O(\log \log n)$ [Chung and Lu 2002].

The high-degree nodes in the power-law graphs of the Waxman model have only very few edges: the highest degree is only 20. Furthermore, as $\lceil n^\gamma \rceil = 3$, the core is really small and so is the cumulative degree. Compared to the other power-law graphs, the high-degree selection strategy does not produce huge benefits but it still outperforms random selection. In practice, one might add high-degree nodes to the set of landmarks until a certain cumulative degree threshold (for example $\sqrt{n}$ or also a threshold value dependent on τ) is reached.

7. APPROXIMATE DISTANCE ORACLE

Compact routing schemes can be seen as the distributed version of approximate distance oracles. In the following, we use the techniques that allowed us to prove an upper bound on the maximum routing table size to prove an upper bound on the space requirements of an approximate distance oracle (Section 7.1). We then trade query time against space: we provide an approximate distance oracle with linear space requirements (Section 7.2).

7.1. Distance Oracle with Stretch 3

We prove the following theorem.

THEOREM 7.1. *Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ be a constant. Assume Equation (1) is satisfied. For random power-law graphs from* $\mathbf{RPLG}(n, \tau)$ *(Definition 3.2), there exists a preprocessing algorithm that runs in expected time $O(n^{1+\gamma} \log n)$ and creates a distance oracle of expected size $O(n^{1+\gamma})$. These bounds also hold with probability at least $1 - 1/n$. After preprocessing, approximate distance queries can be answered in $O(1)$ time with stretch at most* 3.

We propose a modification of the distance oracle by Thorup and Zwick [2005, Fig. 5] for $k = 2$, which guarantees stretch 3. The main idea of the scheme by Thorup and Zwick for $k = 2$ is the following: in the preprocessing step, given a graph $G = (V, E)$, (1) each node $v \in V$ is chosen as a *landmark* independently at random with probability $n^{-1/2}$. The expected number of landmarks is $\sqrt{n}$. (2) For each node $u \in V$, find its nearest landmark $\ell(u)$ and compute the distances from u to all landmarks. To guarantee optimal stretch for short distance queries, (3) for every node $u \in V$ a local *ball* $B_G(u) = \{u' \in V(G) : d(u, u') < d(u, \ell(u))\}$ is computed, including all nodes with distance strictly less than the distance to the landmarks. The result of the distance query $d(s, t)$ is exact if $s \in B(t)$ or $t \in B(s)$ and otherwise stretch 3 is guaranteed [Cowen 2001]. Since the set of landmarks consists of a random sample, the expected ball size is $O(\sqrt{n})$, which is equal to the number of landmarks. This is the optimal balance for general graphs.

For power-law graphs a *better* balance is possible. Using high-degree nodes as landmarks is a natural heuristic. We can select fewer landmarks and obtain smaller sized balls than [Thorup and Zwick 2005, Fig. 5] at the same time.

Details for the preprocessing step are listed in Algorithm 4.

LEMMA 7.2. *Let $\gamma = \frac{\tau-2}{2\tau-3} + \varepsilon$ be a constant. Assume Equation (1) is satisfied. For random power-law graphs* $\mathbf{RPLG}(n, \tau)$*, Algorithm 4 runs in expected time* $O(n^{1+\gamma} \log n)$

ALGORITHM 4: Preprocess $(G = (V, E), \gamma')$

compute core $\leftarrow \{v \in V : \deg(v) > n^{\gamma'}/4\}$
for each $v \in$ core **do**
  run breadth-first search from v in G
  for each node $u \neq v$, store $d(u, v)$
end for
for each $u \in V$ **do**
  compute and store $B_{\text{core}}(u)$ (including distances)
end for

and creates a distance oracle of expected size $O(n^{1+\gamma})$. These bounds also hold with probability at least $1 - 1/n$.

PROOF. The analysis of the compact routing scheme can be applied directly (Lemma 5.11 and proof of Lemma 5.13). □

The query algorithm is the same as in [Thorup and Zwick 2005] for $k = 2$, see Algorithm 5.

ALGORITHM 5: Distance (s, t)

if $s \in B_S(t)$ or $t \in B_S(s)$ **then**
  return local distance $d(s, t)$ from the information at s or t.
else
  return $d(s, \ell(t)) + d(\ell(t), t)$
end if

LEMMA 7.3. *Algorithm 5 runs in time $O(1)$ and achieves stretch 3.*

PROOF. Stretch and time bounds from [Thorup and Zwick 2005] apply. For each node preprocessed information is stored in a hash table with constant access time [Fredman et al. 1984]. □

Theorem 7.1 is immediate from Lemmas 7.2 and 7.3.

7.2. Linear-Space Data Structure

In practical scenarios (such as social networks with millions of individuals) the graph may be too large to store a distance oracle that requires super-linear space. In the following, we propose and analyze a distance data structure that can be stored using linear space. For any subset of $O(\sqrt{n})$ nodes a complete distance table can be stored using linear space. Long-range distances can be approximated by passing through two landmarks in the subset. For short-range distances, we again use balls. These balls can be either pre-computed (resulting in constant query time but larger space requirements) or they can be explored at query time (resulting in longer query time but smaller space requirements).

More generally, we can trade space against query time, depending on the application's needs. For a parameter $\xi \in [0, 1/2]$ we choose a set of landmarks of size $O(n^{1/2+\xi})$ and compute a complete distance table, which requires space $O(n^{1+2\xi})$. Furthermore, for each node $v \in V$ we store its nearest landmark. At query time, given two nodes s and t, we explore both balls (similar to Algorithm 5, but now the balls have not been precomputed). The query time increases, as described by the following theorem.

THEOREM 7.4. *Let $\xi \in [0, 1/2]$ and let $\gamma'' = (1/2-\xi)(1-1/(\tau-1))+\varepsilon$ be two constants. Assume Equation (1) is satisfied. For random power-law graphs from* $\mathbf{RPLG}(n, \tau)$ *(Definition 3.2), there exists a preprocessing algorithm that runs in time $O(n^{3/2+\xi} \log n)$ and creates a distance data structure of size $O(n^{1+2\xi})$. After preprocessing, approximate distance queries can be answered in expected time $O(n^{\gamma''} \log n)$ with stretch at most 5. The bound on the query time also holds with probability at least $1 - 1/n$.*

Note that we may set $\xi := 0$, in which case we obtain a linear-space distance "oracle" with query time $O(n^{\frac{1-1/(\tau-1)}{2}+\varepsilon} \log n)$. For any $\tau \in (2, 3)$ the exponent is at most $1/4 + \varepsilon$. For smaller values of τ (not too far from 2) the exponent is roughly $1/10$.

The remainder of this section is devoted to the proof of Theorem 7.4. Algorithm 6 lists the pseudo-code for the preprocessing algorithm, which is very similar to Algorithm 4.

ALGORITHM 6: Preprocess $(G = (V, E), \xi)$

```
compute core as the set of the n^{1/2+ξ} nodes with highest degree
for each v ∈ core do
   run breadth-first search from v in G
   for each node v ≠ u ∈ core, store d(u, v)
   for each node u ≠ v update ℓ(u) if v is the nearest landmark
end for
```

LEMMA 7.5. *Let $\xi \in [0, 1/2]$ and let $\varepsilon > 0$ be two constants. Assume Equation (1) is satisfied. For random power-law graphs from* $\mathbf{RPLG}(n, \tau)$ *the following holds: with respect to the core with $\Theta(n^{1/2+\xi})$ nodes, each ball has size at most $O(n^{(1/2-\xi)(1-1/(\tau-1))+\varepsilon} \log n)$ with probability at least $1 - 1/n^2$.*

PROOF. We start by estimating the volume of the core. The smallest volume of any node in the core is given by w_k for $k = n^{1/2+\xi}$, which is

$$\begin{aligned} w_k &= \left(\frac{n}{k}\right)^{\frac{1}{\tau-1}} \\ &= n^{\frac{1/2-\xi}{\tau-1}}. \end{aligned}$$

The asymptotic volume of the core is thus at least

$$n^{1/2+\xi} \cdot n^{\frac{1/2-\xi}{\tau-1}},$$

which implies (using Lemma 5.8 in the exact same way as in the proof of Lemma 5.11) that balls have size at most (up to constant and logarithmic factors)

$$n^{1-(1/2+\xi)-\frac{1/2-\xi}{\tau-1}+\varepsilon} = n^{(1/2-\xi)(1-1/(\tau-1))+\varepsilon}.$$

□

The pseudo-code of the query algorithm is listed as Algorithm 7.

ALGORITHM 7: Distance (s, t)

```
explore B_S(s) and B_S(t) using BFS, return distance if s or t are found
if s ∉ B_S(t) and t ∉ B_S(s) then
   return d(s, ℓ(s)) + d(ℓ(s), ℓ(t)) + d(ℓ(t), t)
end if
```

LEMMA 7.6. *Algorithm 7 achieves stretch* 5.

PROOF. Given a query pair (s,t), we have that either $s \in B(t)$ or $t \in B(s)$ (in which case the algorithm returns the exact distance) or

$$\begin{aligned}\tilde{d}(s,t) &= d(s,\ell(s)) + d(\ell(s),\ell(t)) + d(\ell(t),t)\\ &\leq d(s,t) + d(\ell(s),\ell(t)) + d(s,t)\\ &\leq d(s,t) + 3d(s,t) + d(s,t) = 5d(s,t)\end{aligned}$$

□

Lemma 7.5 and Lemma 7.6 imply Theorem 7.4.

Note. Using recent improvements on distance oracles [Patrascu and Roditty 2010; Agarwal et al. 2011], there also is a query algorithm with better stretch, exploiting that we can compute whether the two balls intersect.

8. CONCLUSION

Our analysis provides theoretical justification that high-degree nodes in power-law graphs are indeed very powerful for finding shortest paths in such networks, and thus are effective in improving the performance of shortest-path-related computations.

Perhaps the most intriguing question is whether even polylogarithmic tables would suffice to route with small stretch in power-law graphs. Recent results on distance oracles [Sommer et al. 2009] suggest that $n^{1+\epsilon}$ space is necessary to answer distance queries in constant time for sparse graphs. The lower bound does not extend to routing though. It also remains open whether the scheme by Thorup and Zwick for general k can be optimized for power-law graphs and whether similar techniques can be applied to the name-independent scheme by Abraham et al. [2008]. An average-case analysis of the actual scheme by Thorup and Zwick would be interesting as well as a rigorous analysis of the scheme by Brady and Cowen [2006]. Routing with additive stretch, however, appears to require large routing tables [Gavoille and Sommer 2011]. Furthermore, the analysis for other random power-law graphs models could be an interesting topic.

ACKNOWLEDGMENTS

The second author thanks Michael E. Houle and Mikkel Thorup for their valuable comments on an earlier version of this article. We also thank the anonymous DISC and TALG reviewers for their comments and suggestions.

REFERENCES

ABRAHAM, I., GAVOILLE, C., GOLDBERG, A. V., AND MALKHI, D. 2006a. Routing in networks with low doubling dimension. In *26th International Conference on Distributed Computing Systems (ICDCS)*.

ABRAHAM, I., GAVOILLE, C., AND MALKHI, D. 2005. Compact routing for graphs excluding a fixed minor. In *19th International Symposium on Distributed Computing (DISC)*. 442–456.

ABRAHAM, I., GAVOILLE, C., AND MALKHI, D. 2006b. On space-stretch trade-offs: lower bounds. In *18th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. 207–216.

ABRAHAM, I., GAVOILLE, C., AND MALKHI, D. 2006c. On space-stretch trade-offs: upper bounds. In *18th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. 217–224.

ABRAHAM, I., GAVOILLE, C., MALKHI, D., NISAN, N., AND THORUP, M. 2008. Compact name-independent routing with minimum stretch. *ACM Transactions on Algorithms 4,* 3.

ACHLIOPTAS, D., CLAUSET, A., KEMPE, D., AND MOORE, C. 2009. On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs. *Journal of the ACM 56,* 4. Announced at STOC 2005.

AGARWAL, R., GODFREY, P. B., AND HAR-PELED, S. 2011. Approximate distance queries and compact routing in sparse graphs. In *30th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*.

AIELLO, W., CHUNG, F. R. K., AND LU, L. L. 2000. A random graph model for massive graphs. In *32nd ACM Symposium on Theory of Computing (STOC)*. 171–180.

BARABÁSI, A.-L. AND ALBERT, R. 1999. Emergence of scaling in random networks. *Science 286,* 5439, 509–512.

BOURGAIN, J. 1985. On lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics 52,* 1-2, 46–52.

BRADY, A. AND COWEN, L. 2006. Compact routing on power law graphs with additive stretch. In *9th Workshop on Algorithm Eng. and Exper. (ALENEX)*. 119–128.

CAIDA COOPERATIVE ASSOCIATION FOR INTERNET DATA ANALYSIS. 2003. CAIDA's router-level topology measurements. Online at `http://www.caida.org/tools/measurement/skitter/router_topology/`, file: `itdk0304_rlinks_undirected.gz`.

CHAN, T. M. 2007. More algorithms for all-pairs shortest paths in weighted graphs. In *39th ACM Symposium on Theory of Computing (STOC)*. 590–598.

CHENG, J. AND YU, J. X. 2009. On-line exact shortest distance query processing. In *12th International Conference on Extending Database Technology (EDBT)*. 481–492.

CHUNG, F. R. K. AND LU, L. L. 2002. The average distances in random graphs with given expected degrees. *Internet Mathematics 1*, 91–114. See also PNAS 99, pp. 15879–15882.

CHUNG, F. R. K. AND LU, L. L. 2006. *Complex Graphs and Networks*. American Mathematical Society.

CLAUSET, A., SHALIZI, C. R., AND NEWMAN, M. E. J. 2009. Power-law distributions in empirical data. *SIAM Review 51,* 4, 661–703.

COHEN, E., HALPERIN, E., KAPLAN, H., AND ZWICK, U. 2003. Reachability and distance queries via 2–hop labels. *SIAM Journal on Computing 32,* 5, 1338–1355. Announced at SODA 2002.

COWEN, L. 2001. Compact routing with minimum stretch. *J. Algorithms 38,* 1, 170–183.

CVETKOVSKI, A. AND CROVELLA, M. 2009. Hyperbolic embedding and routing for dynamic graphs. In *28th IEEE International Conference on Computer Communications (INFOCOM)*. 1647–1655.

DIJKSTRA, E. W. 1959. A note on two problems in connection with graphs. *Numerische Math. 1*, 269–271.

DINITZ, M. 2007. Compact routing with slack. In *26th ACM Symposium on Principles of Distributed Computing (PODC)*. 81–88.

ENACHESCU, M., WANG, M., AND GOEL, A. 2008. Reducing maximum stretch in compact routing. In *27th IEEE International Conference on Computer Communications (INFOCOM)*. 336–340.

ERDŐS, P. AND RÉNYI, A. 1960. On the evolution of random graphs. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei 5*, 17–61.

FALOUTSOS, M., FALOUTSOS, P., AND FALOUTSOS, C. 1999. On power-law relationships of the Internet topology. In *SIGCOMM: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication*. 251–262.

FERRANTE, A., PANDURANGAN, G., AND PARK, K. 2008. On the hardness of optimization in power-law graphs. *Theoretical Computer Science 393,* 1-3, 220–230.

FRAIGNIAUD, P. AND GAVOILLE, C. 2001. Routing in trees. In *28th International Colloquium on Automata, Languages and Programming (ICALP)*. 757–772.

FREDMAN, M. L., KOMLÓS, J., AND SZEMERÉDI, E. 1984. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM 31,* 3, 538–544. Announced at FOCS 1982.

GAVOILLE, C. AND HANUSSE, N. 1999. Compact routing tables for graphs of bounded genus. In *26th International Colloquium on Automata, Languages and Programming (ICALP)*. 351–360.

GAVOILLE, C. AND PERENNES, S. 1996. Memory requirements for routing in distributed networks (extended abstract). In *15th Symposium on Principles of Distributed Computing (PODC)*. 125–133.

GAVOILLE, C. AND SOMMER, C. 2011. Sparse spanners vs. compact routing. In *23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*. 225–234.

GOLDBERG, A. V. AND HARRELSON, C. 2005. Computing the shortest path: A* search meets graph theory. In *16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 156–165.

GOLDMAN, R., SHIVAKUMAR, N., VENKATASUBRAMANIAN, S., AND GARCIA-MOLINA, H. 1998. Proximity search in databases. In *24th International Conference on Very Large Data Bases (VLDB)*. 26–37.

KONJEVOD, G., RICHA, A. W., AND XIA, D. 2006. Optimal-stretch name-independent compact routing in doubling metrics. In *25th ACM Symposium on Principles of Distributed Computing (PODC)*. 198–207.

KONJEVOD, G., RICHA, A. W., XIA, D., AND YU, H. 2007. Compact routing with slack in low doubling dimension. In *26th ACM Symposium on Principles of Distributed Computing (PODC)*. 71–80.

KORMAN, A. 2008. Improved compact routing schemes for dynamic trees. In *27th ACM Symposium on Principles of Distributed Computing (PODC)*. 185–194.

KRIOUKOV, D. V., FALL, K. R., AND YANG, X. 2004. Compact routing on internet-like graphs. In *23rd IEEE International Conference on Computer Communications (INFOCOM)*.

KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., SIVAKUMAR, D., TOMKINS, A., AND UPFAL, E. 2000. Random graph models for the web graph. In *41st IEEE Symposium on Foundations of Computer Science (FOCS)*. 57–65.

LI, L., ALDERSON, D., DOYLE, J. C., AND WILLINGER, W. 2005. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics 2,* 4, 431–523.

LU, H.-I. 2002a. Improved compact routing tables for planar networks via orderly spanning trees. In *8th International Computing and Combinatorics Conference*. 57–66.

LU, L. L. 2002b. Probabilistic methods in massive graphs and internet computing. Ph.D. thesis, University of California San Diego.

MCDIARMID, C. 1998. *Probabilistic Methods for Algorithmic Discrete Mathematics*. Algorithms and Combinatorics Series, vol. 16. Springer, Chapter Concentration, 1–46.

MEDINA, A., LAKHINA, A., MATTA, I., AND BYERS, J. W. 2001. Brite: An approach to universal topology generation. In *9th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*. 346–.

MITZENMACHER, M. 2003. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics 1,* 2.

NEWMAN, M. E. J. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics 46*, 323–351.

NEWMAN, M. E. J., STROGATZ, S. H., AND WATTS, D. J. 2001. Random graphs with arbitrary degree distributions and their applications. *Physical Review E 64,* 2, 026118.

NORROS, I. AND REITTU, H. 2006. On a conditionally Poissonian graph process. *Advances in Applied Probability 38,* 1, 59–75.

PAPADOPOULOS, F., KRIOUKOV, D. V., BOGUÑÁ, M., AND VAHDAT, A. 2010. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *29th IEEE International Conference on Computer Communications (INFOCOM)*. 2973–2981.

PATRASCU, M. AND RODITTY, L. 2010. Distance oracles beyond the Thorup-Zwick bound. In *51st IEEE Symposium on Foundations of Computer Science (FOCS)*. 815–823.

PELEG, D. AND UPFAL, E. 1989. A trade-off between space and efficiency for routing tables. *Journal of the ACM 36,* 3, 510–530.

POTAMIAS, M., BONCHI, F., CASTILLO, C., AND GIONIS, A. 2009. Fast shortest path distance estimation in large networks. In *18th ACM conference on Information and knowledge management (CIKM)*. 867–876.

ROUGHAN, M., WILLINGER, W., MAENNEL, O., PEROULI, D., AND BUSH, R. 2011. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications 29,* 9, 1810–1821.

SARMA, A. D., GOLLAPUDI, S., NAJORK, M., AND PANIGRAHY, R. 2010. A sketch-based distance oracle for web-scale graphs. In *3rd International Conference on Web Search and Web Data Mining (WSDM)*. 401–410.

SEN, S. 2009. Approximating shortest paths in graphs. In *3rd International Workshop on Algorithms and Computation (WALCOM)*. 32–43.

SOMMER, C. 2010. Approximate shortest path and distance queries in networks. Ph.D. thesis, The University of Tokyo.

SOMMER, C., VERBIN, E., AND YU, W. 2009. Distance oracles for sparse graphs. In *50th IEEE Symposium on Foundations of Computer Science (FOCS)*. 703–712.

THORUP, M. AND ZWICK, U. 2001. Compact routing schemes. In *13th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. 1–10.

THORUP, M. AND ZWICK, U. 2005. Approximate distance oracles. *Journal of the ACM 52,* 1, 1–24.

WAXMAN, B. M. 1988. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications 6,* 9, 1617–1622.

XIAO, Y., WU, W., PEI, J., WANG, W., AND HE, Z. 2009. Efficiently indexing shortest paths by exploiting symmetry in graphs. In *12th International Conference on Extending Database Technology (EDBT)*. 493–504.

Received October 2010; revised November 2011; accepted April 2012

Online Appendix to: A Compact Routing Scheme and Approximate Distance Oracle for Power-law Graphs[3]

WEI CHEN, Microsoft Research Asia
CHRISTIAN SOMMER, Massachusetts Institute of Technology
SHANG-HUA TENG, University of Southern California
YAJUN WANG, Microsoft Research Asia

A. DETAILS FOR THE BRITE GRAPHS USED IN THE EXPERIMENTS

We provide the detailed parameters used to generate the graphs using BRITE [Medina et al. 2001], based on the Barabási [Barabási and Albert 1999] and Waxman [Waxman 1988] models. We use the prefix of AS to denote the Autonomous System topology and RT to denote the Router Topology.
Model (1 - RTWaxman): 10000 1000 100 1 2 0.15 0.2 1 1 10.0 1024.0
Model (2 - RTBarabasi): 10000 1000 100 1 2 1 10.0 1024.0
Model (3 - ASWaxman): 10000 1000 100 1 2 0.15 0.2 1 1 10.0 1024.0
Model (4 - ASBarabasi): 10000 1000 100 1 2 1 10.0 1024.0

The resulting graphs have the following numbers of nodes and edges, and the corresponding power-law exponent $\hat{\tau}$, estimated using [Newman 2005].

Graph	Nodes	Edges	$\hat{\tau}$
ASWaxman	10,000	20,000	2.806
RTWaxman	10,000	20,000	2.806
ASBarabasi	10,000	19,997	2.893
RTBarabasi	10,000	19,997	2.892

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000